Available online at www.sciencedirect.com
ScienceDirect
Journal of Symbolic Computation 43 (2008) 140–152
Journal of Symbolic Computation
www.elsevier.com/locate/jsc

Algorithms for quadratic forms

Przemysław Koprowski

Instytut Matematyki, Uniwersytet Śląski, ul. Bankowa 14, PL-40-007 Katowice, Poland

Received 15 April 2007; accepted 18 October 2007
Available online 24 October 2007

Abstract

We present algorithms for square classes, quadratic forms and Witt classes of quadratic forms over the field of rational functions of one variable over the reals. The algorithms are capable of: finding the unique representative of a square class, deciding if a given function is a square or a sum of squares and deciding if a quadratic form is isotropic or hyperbolic. Moreover we propose a representation for Witt classes of quadratic forms. With this representation one can manipulate Witt classes without operating directly on their coefficients. We present algorithms both for computing this representation and manipulating Witt classes.

Keywords: Algorithms for square classes of rational functions; Algorithms for quadratic forms; Algorithms for Witt classes

1. Introduction

The algebraic theory of quadratic forms is nowadays rich and well-developed. The main results of the theory may be found in Lam (2005), Scharlau (1985) and Szymiczek (1997). Nevertheless, little has been done so far to develop *algorithms* dealing with central questions of this theory. Existing papers are mainly focused on forms over rational numbers (see e.g. Cremona and Rusin (2003) and Simon (2005b,a)) only recently we have been witnessing the advent of papers dealing with forms over number fields (e.g. Fukshansky (2006)). There are also a few papers dealing with sums of squares of polynomials (see e.g. de Loera and Santos (1996), Powers and Wörmann (1998), Schweighofer (2002) and Reznick (2005)), which is a problem very closely related to the one discussed here. The main purpose of this paper is to present algorithms for quadratic forms over the field $\mathbb{R}(t)$ of rational functions of one variable over the reals.

E-mail address: pkoprowski@member.ams.org.

0747-7171/$ - see front matter
doi:10.1016/j.jsc.2007.10.004

This paper is organized as follows. In Section 3 we deal with the group $\mathbb{R}(t)^*/\mathbb{R}(t)^{*2}$ of square classes of rational functions. For a given non-zero rational function f/g we compute the unique representative $\pm h$ of the square class of f/g, with h a monic square-free polynomial (see Algorithm 1). Other algorithms of this section compute the representative of the product of two square classes (see Algorithm 3), check if a given rational function is a square (Algorithm 2) or a sum of squares (Algorithm 4). The algorithms of Section 3 are subsequently used in Section 4. Here we deal with quadratic forms per se. We present two algorithms. Algorithms 6 and 7 decide respectively if a given quadratic form is isotropic or hyperbolic. We discuss also the problem of finding the anisotropic part of a given isotropic form.

Next, in Section 5 we deal with Witt classes of quadratic forms. Recall that two quadratic forms φ, ψ are said to be *similar* if there exist hyperbolic forms H_1, H_2 such that $\varphi \perp H_1$ is isometric to $\psi \perp H_2$. The equivalence classes induced by this relation are called *Witt classes* of quadratic forms. It is well-known that the Witt classes, with operations induced by an orthogonal sum and a tensor product, form a ring. This ring is called the *Witt ring* of $\mathbb{R}(t)$ and denoted $W\mathbb{R}(t)$. We propose here a representation (see Definition 7) for Witt classes that admits a simple manipulation of these classes without time-consuming calculation on the rational functions. In particular we show how to find a representation of a given quadratic form (see Algorithm 5) and how to compute the sum and the product of two Witt classes having only their representations (Algorithms 8 and 9). We tackle also the problem of reconstructing a representative of a given Witt class (see Algorithm 10).

The algorithms presented in this paper require that the real numbers used as the coefficients of the polynomials are discretely represented (e.g. they are rationals or real algebraic numbers). Moreover, all the algorithms, except Algorithms 8 and 9, remain valid once the field $\mathbb{R}$ is substituted by an arbitrary real closed field R. At the end of the paper we comment on how to alter the two remaining algorithms to suit this more general context. For reader's convenience we gather all the algorithms in the very last section.

2. Notation

We adopt a standard notation used in the field of quadratic forms as it exists in Lam (2005), Scharlau (1985) and Szymiczek (1997). We use Latin letters f, g, h to denote both: polynomials and (by abuse of notation) their square classes. Next, the symbol $\langle f_1, \ldots, f_N \rangle$ means a quadratic form $f_1 X_1^2 + \cdots + f_N X_N^2$ defined over $\mathbb{R}(t)$. All the forms appearing in this paper are assumed to be nonsingular. The orthogonal sum and the tensor product of quadratic forms are denoted respectively by $\perp$ and $\otimes$. For a quadratic form $\varphi = \langle a_1, \ldots, a_N \rangle$ we denote by $\operatorname{sgn}\varphi$ the signature of the form:

$$\operatorname{sgn}\varphi := \sum_{i=1}^{N} \operatorname{sgn}(a_i).$$

Finally, the leading coefficient of a polynomial f is denoted $\operatorname{lc}(f)$.

3. Algorithms for the square-class group of $\mathbb{R}(t)$

Before we proceed to discuss algorithms for quadratic forms, we first present algorithms dealing with their "atomic components", namely the square classes of $\mathbb{R}(t)$. Take a non-zero rational function f/g. Its square class $(f/g)\cdot\mathbb{R}(t)^{*2}$ contains the unique square-free polynomial $h \in \mathbb{R}[t]$ with the leading coefficient equal to either 1 or -1. By the analogy to monic polynomials, we shall call such polynomials *semi-monic*. Our first task is to determine h. This is trivial if we

can factor f, g into linear terms. Unfortunately, it is not possible in general. Instead we may use square-free factorization (see e.g. Geddes et al. (1992), Chapter 8.2). Recall that a square-free factorization algorithm for a given polynomial f of degree n returns square-free monic polynomials $h_1, \ldots, h_n$ such that the roots of h_i are precisely the roots of f of multiplicity i. In particular

$$f = \mathrm{lc}(f) \cdot h_1 \cdot h_2^2 \cdots h_n^n.$$

We have an immediate consequence:

Corollary 1. *Let $f/g \in \mathbb{R}(t)$ be a non-zero rational function. Take $\hat{f} = \frac{1}{\mathrm{lc}(fg)}(fg)$ and let $h_1, \ldots, h_n$ be the output of the square-free factorization of $\hat{f}$. Then the unique semi-monic representative of the square class of f/g is*

$$h = \mathrm{sgn}(\mathrm{lc}(fg)) \cdot h_1 \cdot h_3 \cdots h_{n'},$$

where n' is the biggest odd number less than or equal to n.

We may write down two more observations:

Remark 2. A rational function f/g is a square in $\mathbb{R}(t)$ if and only if the unique semi-monic representative h of its square class returned by Algorithm 1 equals 1.

Remark 3. If f, g are the semi-monic square-free representatives of two square classes, then the semi-monic square-free representative of their product is $(fg)/\big(\gcd(f, g)\big)^2$.

For reader's convenience we explicitly write down Algorithms 1–3 implied by the above observations. Another natural question is whether a given rational function is a sum of squares in $\mathbb{R}(t)$. In other word, if the form $\langle -f, 1, 1\rangle$ is isotropic. We deal with the problem of isotropy in the next section, however, in this particular situation, one does not actually need to use the general algorithm presented there. In fact much simpler solution follows from the Artin's Theorem (see e.g Lam (2005), Theorem VIII.1.12): f is a sum of squares if and only if it is non-negative everywhere. Since we may assume that f is semi-monic square-free, this condition boils down to checking if the leading coefficient of f equals one and f has no real roots. The last condition can be checked using the Sturm sequence (see Basu et al. (2003), Algorithm 9.28). The above discussion is summarized in Algorithm 4.

4. Algorithms for quadratic forms

We now turn our attention to quadratic forms themselves. Take a quadratic form φ over $\mathbb{R}(t)$. The Gram–Schmidt algorithm for orthogonalization of a quadratic form is well-known. Thus, we may assume that $\varphi = \langle f_1, \ldots, f_N\rangle$ is already in a diagonal form and $f_1, \ldots, f_N$ are semi-monic square-free. The two most natural questions are: *Is φ isotropic? Is φ hyperbolic?*

First we address the question of isotropy. Unary forms cannot be isotropic. A binary form $\langle f, g\rangle$ is isotropic if and only if it is hyperbolic if and only if — fg is a square — we have already presented Algorithm 2 to decide this. This leaves us with the problem of deciding when a form of dimension $N \geq 3$ is isotropic. To this end we utilize a version of Strong Hasse Principle known as Witt theorem — see Witt (1998, Satz 22, page 14) or Knebusch (1976, Theorem 9.4). The direct consequence of it is the following corollary.

Corollary 4. *Let $\varphi = \langle f_1, \ldots, f_N\rangle$ be a quadratic form over $\mathbb{R}(t)$ with $N \geq 3$. Assume that all f_i $(1 \leq i \leq N)$ are semi-monic square-free. Let further $\mathfrak{p}_1 < \cdots < \mathfrak{p}_s$ be all the roots of all f_i's*

and denote $\mathfrak{p}_0 := -\infty$, $\mathfrak{p}_{s+1} := +\infty$. *For every* $0 \leq i \leq s$ *fix* $\mathfrak{q}_i \in (\mathfrak{p}_i, \mathfrak{p}_{i+1})$. *Finally take*

$$l := \max\bigl\{|\operatorname{sgn}\varphi(\mathfrak{q}_i)| : 0 \leq i \leq s\bigr\}.$$

Then φ *is isotropic if and only if* $l < N$.

Thus the only thing, we need to know in order to construct an algorithm, deciding if a given quadratic form is isotropic, is a method of finding the signs of f_i's in some intermediate points between their roots. Fortunately, the solution is well-known (see e.g. Basu et al. (2003), Chapter 10). Let $D = f_1 \cdots f_N$ denote the determinant of φ and $d = D/\gcd(D, D')$ be the square-free polynomial with all the same roots as D, then:

- all the roots of d are precisely $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ and each is of the multiplicity one;
- the derivative d' has a root in each interval $(\mathfrak{p}_i, \mathfrak{p}_{i+1})$ for $1 \leq i < s$.

Now the algorithm `UnivariateSignDetermination` (see Basu et al. (2003), Algorithm 10.61) determines the sign of each f_i in every root of d'. The two remaining intervals, namely $(\mathfrak{p}_0, \mathfrak{p}_1)$ and $(\mathfrak{p}_s, \mathfrak{p}_{s+1})$, are the neighborhoods of infinity. Hence, one needs only to examine the signs of the leading coefficients of f_i's. All in all, we have just proved the correctness of Algorithms 5 and 6. Algorithm 5 returns more information than needed by Algorithm 6 (namely the variables d, D, n in the notion of the algorithm). Its usage will become clear in the next section.

We may now present an algorithm deciding if a given form is hyperbolic or equivalently if two forms belong to the same Witt class (see below). Here a version of *Weak* Hasse Principle (see e.g. Knebusch (1976), Theorem 9.5) due again to Witt (see Witt (1998), Satz 23, page 14) provides us with the following criterion:

Proposition 5. *Let* $\varphi = \langle f_1, \ldots, f_N\rangle$ (N *even*) *be a quadratic form over* $\mathbb{R}(t)$. *Assume that all* f_i ($1 \leq i \leq N$) *are semi-monic square-free. Let further* $\mathfrak{p}_1 < \cdots < \mathfrak{p}_s$ *be all the roots of all* f_i*'s and denote* $\mathfrak{p}_0 := -\infty$, $\mathfrak{p}_{s+1} := +\infty$. *For every* $0 \leq i \leq s$ *fix* $\mathfrak{q}_i \in (\mathfrak{p}_i, \mathfrak{p}_{i+1})$. *Then* φ *is hyperbolic if and only if* $\operatorname{sgn}(\mathfrak{q}_i) = 0$ *for every* $0 \leq i \leq s$ *and* $(-1)^{N/2} f_1 \cdots f_N$ *is a square.*

The proof follows immediately from the Witt theorem (see Witt (1998), Satz 23, page 14). Algorithm 7 is thus similar to the previous one.

Recall (see e.g. Szymiczek (1997), Chapter 12.1) that every isotropic form φ can be decomposed into

$$\varphi = \psi \perp \eta,$$

with η hyperbolic and ψ anisotropic. The form ψ is called the *anisotropic part* of φ and is uniquely determined up to isometry. There are two interesting related problems. If a given form φ is isotropic, how to find the anisotropic part of φ? How to find an isotropic vector? The following example shows that in general it is impossible to answer the first question without knowing the roots of the polynomials involved. Thus, we cannot hope to find an algorithm solving the first problem by manipulating only the coefficients of the polynomials (on the other hand, it is possible to find the anisotropic part in terms of roots of the determinant, see Algorithm 10).

Example 6. Take a monic unsolvable polynomial $f \in \mathbb{Q}[t]$ (i.e. the roots of f cannot be expressed by radicals) having only real roots $\mathfrak{p}_1 < \mathfrak{p}_2 < \cdots < \mathfrak{p}_n$ (necessarily $n \geq 5$) and let $\mathfrak{q} \in (\mathfrak{p}_1, \mathfrak{p}_2)$ be a rational number. For example

$$f = t^5 - 5t^3 + 4t - 1 \quad \text{and} \quad \mathfrak{q} = -\frac{3}{2}.$$

Consider the form φ defined by

$$\varphi = \langle 1, t-\mathfrak{q}, f, -(t-\mathfrak{q})\cdot f\rangle.$$

The form is isotropic. Compute the local signatures

	$(-\infty, \mathfrak{p}_1)$	$(\mathfrak{p}_1, \mathfrak{q})$	$(\mathfrak{q}, \mathfrak{p}_2)$	$(\mathfrak{p}_2, \mathfrak{p}_3)$	$(\mathfrak{p}_3, \mathfrak{p}_4)$	$(\mathfrak{p}_4, \mathfrak{p}_5)$	$(\mathfrak{p}_5, \infty)$
1	$+$	$+$	$+$	$+$	$+$	$+$	$+$
$t-\mathfrak{q}$	$-$	$-$	$+$	$+$	$+$	$+$	$+$
f	$-$	$+$	$+$	$-$	$+$	$-$	$+$
$-(t-\mathfrak{q})\cdot f$	$-$	$+$	$-$	$+$	$-$	$+$	$-$
φ	-2	2	2	2	2	2	2

It is clear that $\langle t-\mathfrak{p}_1, t-\mathfrak{p}_1\rangle$ is the anisotropic part of φ, but $\mathfrak{p}_1$ cannot be expressed by radicals! We claim that for any two square-free polynomials $g, h \in \mathbb{R}[t]$, if the form $\langle g, h\rangle$ is isometric to $\langle t-\mathfrak{p}_1, t-\mathfrak{p}_1\rangle$, then the coefficients of g, h cannot be expressed in terms of radicals, either. Indeed, comparing signatures, we see that both g, h must change signs precisely at $\mathfrak{p}_1$ and nowhere else. Thus, $\mathfrak{p}_1$ is the only real root of both of them. Now, since, all the roots of f are real, we have

$$t-\mathfrak{p}_1 = \gcd(f, g).$$

Therefore, the coefficients of g cannot be expressed by radicals.

This example does not preclude the possibility that an algorithmic solution of the second problem exists. However, the two problems seems to be interrelated and so solving the second one may be highly nontrivial.

5. Algorithms for the Witt ring of $\mathbb{R}(t)$

We are now ready to go a step further and consider Witt classes of quadratic forms. Since every non-zero Witt class contains the unique (up to isometry) anisotropic form, the most convenient way of representing a Witt class would be to store this anisotropic representative. This boils down to finding the anisotropic part of a given form. The example at the end of the previous section shows, however, that this cannot be done without knowing the roots of the polynomials in the diagonalization. Thus, we propose here a representation which is easier to manipulate than storing the whole diagonalization of *some* (possibly isotropic) representative. The mentioned example shows that this representation is not worse than storing *some* representative in the sense that in both cases, to reconstruct the anisotropic representative, one must work with the roots of the polynomials involved, not just their coefficients. Algorithm 5 discussed in the previous section provides us with the following convenient representation for the Witt class of a quadratic form.

Definition 7. Given a quadratic form $\varphi = \langle f_1, \ldots, f_N\rangle$ over $\mathbb{R}(t)$, where $f_1, \ldots, f_N$ are semi-monic square-free, we define the *representation* for the Witt class of φ to be a tuple (d, D, n, S) consisting of:

- the monic square-free polynomial d having roots precisely at the roots of f_i's:

$$\bigwedge_{\mathfrak{p}\in\mathbb{R}} \Big(d(\mathfrak{p}) = 0 \iff \bigvee_{1\le i\le N} f_i(\mathfrak{p}) = 0 \Big);$$

- the semi-monic square-free representative D of the signed determinant of φ;
- the remainder $n = N \pmod 4$ of the dimension of φ modulo 4;

- the finite sequence $S = (\sigma_0, \ldots, \sigma_s)$ of the signatures of φ in intervals between consecutive roots of d (with a convention that σ_0 is the signature of φ in a right neighborhood of $-\infty$ and σ_s is the signature of φ in a left neighborhood of $+\infty$).

We show that this representation admits fast and simple addition and multiplication of Witt classes. First, however, let us explicitly formulate the following immediate consequence of Weak Hasse Principle (see Witt (1998), Satz 23, page 14):

Proposition 8. *A tuple (d, D, n, S) represents the zero element of the Witt ring $W\mathbb{R}(t)$ (i.e. the class of a hyperbolic form) if and only if $S = (0, 0, \ldots, 0)$ and $D = 1$. If this is the case then also n is even.*

In particular the representation of a given class is not unique.

Proposition 9. *Let $\varphi = \langle f_1, \ldots, f_N\rangle$, $\psi = \langle g_1, \ldots, g_M\rangle$ be two quadratic forms and let (d_1, D_1, n_1, S_1) and (d_2, D_2, n_2, S_2) denote representations of their Witt classes. Define (d, D, n, S) as follows:*

(a) $d := (d_1 d_2)/\gcd\big(d_1 d_2, (d_1 d_2)'\big)$;
(b) *D is the semi-monic square-free representative of $(-1)^{n_1 n_2} D_1 D_2$;*
(c) $n := (n_1 + n_2) \pmod 4$;
(d) *$S = (\sigma_0, \ldots, \sigma_s)$ is such that in every sample point $\mathfrak{q}_i$ the signature σ_i is the sum of the corresponding signatures from S_1 and S_2.*

Then (d, D, n, S) represents the Witt class of $\varphi \perp \psi$.

Proof. The points (a) and (c) are straightforward. It is also obvious that at every point $\mathfrak{q}$, not being a root of any f_i or g_i, the signature of $\varphi \perp \psi = \langle f_1, \ldots, f_N, g_1, \ldots, g_M\rangle$ is the sum of signatures of φ and ψ at $\mathfrak{q}$. Hence, the only thing left is to prove the formula in (b). Denote by $\hat{D} := f_1 \cdots f_N \cdot g_1 \cdots g_M$ the determinant of $\varphi \perp \psi$. Likewise, let $\hat{D}_1 := f_1 \cdots f_N$ and $\hat{D}_2 := g_1 \cdots g_M$ be the determinants of φ and ψ respectively. Therefore we have $\hat{D} = \hat{D}_1 \cdot \hat{D}_2$. Compute the signed determinants:

$$D = (-1)^{\frac{1}{2}(n_1+n_2)(n_1+n_2-1)} \hat{D}_1 \hat{D}_2, \quad D_1 = (-1)^{\frac{1}{2}n_1(n_1-1)} \hat{D}_1, \quad D_2 = (-1)^{\frac{1}{2}n_2(n_2-1)} \hat{D}_2.$$

Substituting $\hat{D}_1$, $\hat{D}_2$ into the the first equation we arrive at

$$\begin{aligned} D &= (-1)^{\frac{1}{2}(n_1+n_2)(n_1+n_2-1)} \cdot (-1)^{-\frac{1}{2}n_1(n_1-1)} D_1 \cdot (-1)^{-\frac{1}{2}n_2(n_2-1)} D_2 \\ &= (-1)^{n_1 n_2} D_1 D_2 \quad \square \end{aligned}$$

In a similar fashion we may represent the Witt class of the tensor product of two quadratic forms.

Proposition 10. *Let $\varphi = \langle f_1, \ldots, f_N\rangle$, $\psi = \langle g_1, \ldots, g_M\rangle$ be two quadratic forms and let (d_1, D_1, n_1, S_1) and (d_2, D_2, n_2, S_2) denote representations of their Witt classes. Define (d, D, n, S) as follows:*

(a) $d := (d_1 d_2)/\gcd(d_1 d_2, (d_1 d_2)')$;
(b) *D is the semi-monic square-free representative of:*
- *1 if $(n_1, n_2) \in \big\{(0, 0), (2, 2)\big\}$;*
- *D_1 if $(n_1, n_2) \in \big\{(0, 1), (0, 2), (0, 3), (2, 1), (2, 3)\big\}$;*
- *D_2 if $(n_1, n_2) \in \big\{(1, 0), (2, 0), (3, 0), (1, 2), (3, 2)\big\}$;*

- $D_1 D_2$ *if* $(n_1, n_2) \in \{(1,1),(1,3),(3,1),(3,3)\}$;

(c) $n := (n_1 \cdot n_2) \pmod 4$;

(d) $S = (\sigma_0, \ldots, \sigma_s)$ *is such that in every sample point* $\mathfrak{q}_i$ *the signature* σ_i *is the product of the corresponding signatures from* S_1 *and* S_2.

Then (d, D, n, S) *represents the Witt class of* $\varphi \otimes \psi$.

The proof is fully analogous to the previous one. From this two propositions follow the algorithms for computing the sum and the product of two Witt classes. Assume that (d_1, D_1, n_1, S_1) and (d_2, D_2, n_2, S_2) represent the Witt classes of two quadratic forms φ, ψ. Take d, D, n as defined in Proposition 9 (respectively 10). The values of d, D and n are explicitly given, hence we must only compute S. For a root $\mathfrak{p}$ of d let $q_k, q_{k+1} \in \mathbb{Q}$ be two rational numbers such that $\mathfrak{p}$ is the only root of d in the interval (q_k, q_{k+1}). They may be obtained using the algorithm `RealRootIsolation` (cf. Basu et al. (2003), Algorithm 10.41). Now, since d_1 is square-free, $\mathfrak{p}$ is the root of d_1 if and only if d_1 changes sign at $\mathfrak{p}$. This happens if and only if $d_1(q_k) \cdot d_1(q_{k+1}) < 0$. Likewise for the polynomial d_2.

Consequently, in order to construct the list of local signatures of $\varphi \perp \psi$ and $\varphi \otimes \psi$, we proceed as follows. Let $S_1 = (\xi_0, \ldots, \xi_m)$ and $S_2 = (\zeta_0, \ldots, \zeta_m)$. We start from the neighborhood of $-\infty$. The signature of $\varphi \perp \psi$ equals $\xi_0 + \zeta_0$ and the signature of $\varphi \otimes \psi$ equals $\xi_0 \cdot \zeta_0$. Now, we advance toward the neighborhood of $+\infty$, moving by one root of d in each step. In every interval visited, the resulting signature equals $\xi_i + \zeta_j$ for $\varphi \perp \psi$ and $\xi_i \cdot \zeta_j$ for $\varphi \otimes \psi$. The index i (resp. j) is incremented each time we step over the root of d_1 (resp. d_2).

The above two propositions imply the following corollaries:

Corollary 11. *Let* φ, ψ *be two quadratic forms and let* (d_1, D_1, n_1, S_1), (d_2, D_2, n_2, S_2) *denote representations of the Witt classes of* φ *and* ψ. *Then Algorithms 8 and 9 return representations of the Witt classes of* $\varphi \perp \psi$ *and* $\varphi \otimes \psi$ *respectively.*

Corollary 12. *If* (d, D, n, S) *represents the Witt class of a quadratic form* φ, *then the tuple* $(d, (-1)^n D, n, -S)$ *represents the Witt class of* $-\varphi$.

Corollary 13. *Let* (d_1, D_1, n_1, S_1) *and* (d_2, D_2, n_2, S_2) *be two representations of elements of the Witt ring* $W\mathbb{R}(t)$. *They represent the same element if and only if* $n_1 \equiv n_2 \pmod 2$, $D_1 = D_2$ *and Algorithm 8 returns* $(*, 1, *, (0, \ldots, 0))$ *for the sum of* (d_1, D_1, n_1, S_1) *and* $(d_2, (-1)^{n_2} D_2, n_2, -S_2)$.

One may wish to know the anisotropic representative for a computed Witt class. The example at the end of Section 4 shows that in general it is impossible to find it without knowing the roots of the polynomials involved. On the other hand, if we allow the roots of d in the expression, the task becomes manageable. Let (d, D, n, S) be a representation of some Witt class. We shall consider three cases. First, assume that $S = (0, 0, \ldots, 0)$ and so n is even. If $D = 1$ then this is the class of a hyperbolic form and so it does not have the anisotropic representative (see Proposition 8). If $D \neq 1$ then either D or $-D$ is a sum of squares and it follows from Weak Hasse Principle that the class contains the form $\langle D, -\mathrm{lc}(D)\rangle$. Next, assume that every entry in S is either 1 or -1. Using Weak Hasse Principle again we see that the class contains the unary form $\langle D\rangle$.

Consider now the general case. Let $\nu \geq 2$ be the maximum of absolute values of entries in S. Further let $\mathfrak{p}_1 < \cdots < \mathfrak{p}_s$ be all the real roots of d. For $0 \leq i \leq s$ find $d_i, e_i \in \mathbb{Z}$ such that

$$\begin{cases} d_i + e_i = \nu \\ d_i - e_i = \sigma_i. \end{cases}$$

Construct a table A consisting of ν rows and $s+1$ columns and fill it with ± 1 in such a way that $+1$ appears exactly d_i times in the ith column ($0 \leq i \leq s$). The polynomials g_j for $1 \leq j \leq \nu$ can now be formed as

$$g_j := A_{j,s} \cdot \prod_{i=1}^{s} (t - \mathfrak{p}_i)^{\eta_{j,i}},$$

where $A_{j,s}$ is the last entry of the jth row of A equal to the sign of the leading coefficient of g_j; $\eta_{i,j}$ is either zero if the entries $A_{j,i-1}$ and $A_{j,i}$ agree or one if they differ. Finally, take

$$\Lambda := (-1)^{\frac{1}{2}\nu(\nu-1)} \cdot D \cdot g_1 \cdots g_\nu$$

and consider the form $\langle \Lambda g_1, g_2, \ldots, g_\nu \rangle$. It is clear that the signed determinant of φ equals D. Moreover the construction of g_j's ensures the equalities of all local signatures. It follows that φ is the anisotropic representative of the Witt class considered. The above discussion is summarized in Algorithm 10.

6. Examples

The algorithms described in this paper were implemented in a computer algebra system Mathematica 3.01 (Wolfram Research, Inc., 1996). Here we present some example computations. Consider two forms: $\varphi = \langle -t, -t(t^5-t-1) \rangle$ and $\psi = \langle t^5-1, -t^4+1, t-1 \rangle$. Using Algorithm 6 we verify that both forms are anisotropic. Computing their tensor product directly we arrive at the form:

$$\begin{aligned} \varphi \otimes \psi \; = \big\langle & -t^6 + t, t^5 - t, -t^2 + t, -t^{11} + t^7 + 2t^6 - t^2 - t, \\ & t^{10} - 2t^6 - t^5 + t^2 + t, -t^7 + t^6 + t^3 - t \big\rangle. \end{aligned}$$

Using Algorithm 6 again we check that $\varphi \otimes \psi$ is isotropic. Compute the Witt classes of φ and ψ using Algorithm 5:

$$\begin{aligned} \varphi \; & \sim \big((t^5 - t - 1) \cdot t, \; -t^5 + t + 1, \; 2, \; (0, 0, -2) \big) \\ \psi \; & \sim \big((t^3 + t^2 + t + 1)(t^5 - 1), \\ & \qquad (t-1)(t^7 + 2t^6 + 3t^5 + 4t^4 + 4t^3 + 3t^2 + 2t + 1), \; 3, \; (-3, -1, 1) \big). \end{aligned}$$

Now using Algorithm 9 we multiply these classes in the Witt ring and we get

$$\begin{aligned} W_1 := (d, D, n, S) \; = \big(& (t^5 - t - 1)(t^5 - 1)(t^3 + t^2 + t + 1)t, \\ & -t^5 + t + 1, 2, (0, 0, 0, 0, -2) \big). \end{aligned}$$

Hence indeed the product $\varphi \otimes \psi$ is isotropic. Using Algorithm 10 we find the anisotropic representative of this class (hence the anisotropic part of $\varphi \otimes \psi$):

$$\eta = \langle -(t^5 - t - 1), -1 \rangle.$$

The representation of Witt class of η computed by Algorithm 5 is

$$W_2 := \big(t^5 - t - 1, -t^5 + t + 1, 2, (0, -2) \big).$$

Using Corollary 13 we verify that indeed W_1 and W_2 represent the same class.

Final remark

All the presented algorithms, except Algorithms 8 and 9, remain valid also over $R(t)$, where R is an arbitrary real closed field. In order to make these two algorithms work in this more general setup, one needs to use Thoms encodings (see Basu et al. (2003), Section 10.4), instead of rational numbers, to isolate the roots of D.

7. Algorithms

Below we gather all the algorithms presented in this paper.

Algorithm 1: SquareClassRepresentative

Input: $f/g \in \mathbb{R}(t)$ a non-zero rational function
Output: $h \in \mathbb{R}[t]$ the semi-monic representative of the square class of f/g

$s \leftarrow \mathrm{lc}(fg)$;
// Use (Geddes et al., 1992, Algorithm 8.2)
$H = (h_1, \ldots, h_n) \leftarrow$ `SquareFreeFactorization`$(\frac{1}{s} fg)$;
$h \leftarrow \mathrm{sgn}(s) \cdot \prod_{\substack{1 \le i \le n \\ i \notin 2\mathbb{Z}}} h_i$;
return h;

Algorithm 2: IsSquare

Input: $f/g \in \mathbb{R}(t)$ a rational function
Output: **true** if $f/g \in (\mathbb{R}(t))^2$, **false** otherwise

if `SquareClassRepresentative`$(fg) = 1$ **then**
  return true;
else
  return false;

Algorithm 3: MultiplySquareClasses

Input: f, g semi-monic square-free representatives of two square classes
Output: h the semi-monic square-free representative of the square class of fg

return $(f \cdot g)/\big(\gcd(f, g)^2\big)$;

Algorithm 4: IsSumOfSquares

Input: $f/g \in \mathbb{R}(t)$
Output: **true** if f/g is a sum of squares in $R(t)$, **false** otherwise

$h \leftarrow$ `SquareClassRepresentative`(f/g);
if $\mathrm{lc}(h) \neq 1$ **then**
  return false;
// Use the Sturm sequence (Basu et al., 2003, Algorithm 9.28)
$r \leftarrow$ `SturmQuerry`(h);
if $r = 0$ **then**
  return true;
else
  return false;

Algorithm 5: LocalSignatures

Input: $f_1, \ldots, f_N \in R[t]$ semi-monic square-free polynomials
Output: (d, D, n, S) a representation of the Witt class of $\langle f_1, \ldots, f_N \rangle$

$d \leftarrow f_1 \ldots f_N$;
// Compute the remainder of the dimension
$n \leftarrow N \pmod 4$;
// Compute the signed determinant using Algorithm 1
$D \leftarrow (-1)^{\frac{1}{2}n(n-1)} \cdot \texttt{SquareClassRepresentative}(d)$;
// Compute the monic square-free determinant
$d \leftarrow d/\gcd(d, d')$;
$d \leftarrow \frac{1}{\mathrm{lc}(d)} d$;
// Compute the signature near minus infinity
$\sigma_0 \leftarrow \sum_{i=1}^{N} \operatorname{sgn}\big((-1)^{\deg f_i} \mathrm{lc}(f_i)\big)$;
$S \leftarrow (\sigma_0)$;
// If all polynomials f_i are constant we are done
if $\deg d \neq 0$ **then**
  // Use (Basu et al., 2003, Algorithm 10.61)
  $U = (\xi_1, \ldots, \xi_{s-1}) \leftarrow \texttt{UnivariateSignDetermination}(\{f_1, \ldots, f_N, d\}, d')$;
  // Compute the signature near infinity
  $\sigma_s \leftarrow \sum_{i=1}^{N} \operatorname{sgn} \mathrm{lc}(f_i)$;
  // Store exactly one signature for every interval
  $\xi_{0,N+1} \leftarrow (-1)^{\deg d}$;
  $S \leftarrow S \cup \big(\sum_{j=1}^{N} \xi_{i,j} : 1 \leq i < s, \xi_{i,N+1} \neq \xi_{i-1,N+1}\big)$;
  if $\xi_{s-1,N+1} \neq 1$ **then**
    $S \leftarrow S \cup (\sigma_s)$;
return (d, D, n, S);

Algorithm 6: IsIsotropic

Input: $f_1, \ldots, f_N \in R[t]$ semi-monic square-free polynomials
Output: **true** if $\langle f_1, \ldots, f_N \rangle$ is isotropic, **false** otherwise

// Unary forms are anisotropic
if $N = 1$ **then**
  return false;
// For a binary form check the determinant
if $N = 2$ **then**
  return $\texttt{IsIsquare}(-f_1 f_2)$;
// For dimension ≥ 3 use Corollary 4
if $N \geq 3$ **then**
  $(d, D, n, S) \leftarrow \texttt{LocalSignatures}(f_1, \ldots, f_N)$;
  $l \leftarrow \max\{|\sigma| : \sigma \in S\}$;
  if $l < N$ **then**
    return true;
  else
    return false;

Algorithm 7: IsHyperbolic

Input: $f_1, \ldots, f_N \in R[t]$ semi-monic square-free polynomials
Output: **true** if $\langle f_1, \ldots, f_N \rangle$ is hyperbolic, **false** otherwise

// Check if the dimension is even
if $N \equiv 1 \pmod 2$ **then**
  return false;
// Use Proposition 5
$(d, D, n, S) \leftarrow \texttt{LocalSignatures}(f_1, \ldots, f_N)$;
if $S = (0, \ldots, 0) \wedge D = 1$ **then**
  return true;
else
  return false;

Algorithm 8: SumOfWittClasses

Input: $\big(d_1, D_1, n_1, S_1 = (\xi_0, \ldots, \xi_{m_1})\big)$, $\big(d_2, D_2, n_2, S_2 = (\zeta_0, \ldots, \zeta_{m_2})\big)$ representations of the Witt classes of two quadratic forms
Output: (d, D, n, S) a representation of the Witt class of their sum

$d \leftarrow d_1 \cdot d_2$;
$d \leftarrow d / \gcd(d, d')$;
// Compute the semi-monic square-free signed determinant
// using Algorithm 3
$D \leftarrow (-1)^{n_1 n_2} \cdot \texttt{MultiplySquareClasses}(D_1, D_2)$;
$n \leftarrow (n_1 + n_2) \pmod 4$;
$S \leftarrow \emptyset$;
// Isolate roots of d using (Basu et al., 2003, Algorithm 10.41)
$Q = (q_0, \ldots, q_N) \leftarrow \texttt{RealRootIsolation}(d)$;
$i \leftarrow 0$;
$j \leftarrow 0$;
// Compute local signatures
for $k \leftarrow 0$ **to** N **do**
  $S \leftarrow S \cup (\xi_i + \zeta_j)$;
  if $k < N \wedge d_1(q_k) \cdot d_1(q_{k+1}) < 0$ **then**
    $i \leftarrow i + 1$;
  if $k < N \wedge d_2(q_k) \cdot d_2(q_{k+1}) < 0$ **then**
    $j \leftarrow j + 1$;
return (d, D, n, S)

Algorithm 9: ProductOfWittClasses

Input: $\big(d_1, D_1, n_1, S_1 = (\xi_0, \ldots, \xi_{m_1})\big)$, $\big(d_2, D_2, n_2, S_2 = (\zeta_0, \ldots, \zeta_{m_2})\big)$ representations of the Witt classes of two quadratic forms
Output: (d, D, n, S) a representation of the Witt class of their product
$d \leftarrow d_1 \cdot d_2$;
$d \leftarrow d / \gcd(d, d')$;

```
// Compute the semi-monic square-free signed determinant
if (n_1, n_2) ∈ {(0,0), (2,2)} then D ← 1;
else if (n_1, n_2) ∈ {(0,1), (0,2), (0,3), (2,1), (2,3)} then D ← D_1;
else if (n_1, n_2) ∈ {(1,0), (2,0), (3,0), (1,2), (3,2)} then D ← D_2;
else D ← MultiplySquareClasses(D_1, D_2);
n ← (n_1 n_2) (mod 4);
S ← ∅;
// Isolate roots of d using (Basu et al., 2003, Algorithm 10.41)
Q = (q_0, ..., q_N) ← RealRootIsolation(d);
i ← 0;
j ← 0;
// Compute local signatures
for k ← 0 to N do
    S ← S ∪ (ξ_i · ζ_j);
    if k < N ∧ d_1(q_k) · d_1(q_{k+1}) < 0 then
        i ← i + 1;
    if k < N ∧ d_2(q_k) · d_2(q_{k+1}) < 0 then
        j ← j + 1;
return (d, D, n, S)
```

Algorithm 10: AnisotropicRepresentative

```
Input: (d, D, n, S = (σ_0, ..., σ_s)) a representation of a Witt class
Output: the anisotropic representative of the class if (d, D, n, S) represents a non-zero
        class, ⟨1, −1⟩ otherwise
ν ← max{|σ_i| : 1 ≤ i ≤ s};
// Special cases
if ν = 0 then
    return ⟨D, −lc(D)⟩;
if ν = 1 then
    return ⟨D⟩;
// General case
// Construct A
for i ← 0 to s do
    d_i ← (ν + σ_i)/2;
    e_i ← (ν − σ_i)/2;
    for j ← 1 to ν do
        if j ≤ d_i then
            A_{j,i} ← 1;
        else
            A_{j,i} ← −1;
A ← (A_{j,i})_{1≤j≤ν, 0≤i≤s};
// Enumerate real roots of d
```

(*continued on next page*)

$(\mathfrak{p}_1 < \mathfrak{p}_2 < \cdots < \mathfrak{p}_2) \leftarrow \texttt{RealRoots}(d)$;
// Construct the anisotropic form
for $j \leftarrow 1$ **to** ν **do**
  $g_j \leftarrow A_{j,s} \cdot \prod_{1 \le i \le s} (t - \mathfrak{p}_i)^{\frac{1}{2}|A_{j,i} - A_{j,i-1}|}$;
$\Lambda \leftarrow (-1)^{\frac{1}{2}\nu(\nu-1)} \cdot D \cdot g_1 \cdots g_\nu$;
return $\langle \Lambda \cdot g_1, g_2, \ldots, g_\nu \rangle$

References

Basu, S., Pollack, R., Roy, M.-F., 2003. Algorithms in real algebraic geometry. In: Algorithms and Computation in Mathematics, vol. 10. Springer-Verlag, Berlin.

Cremona, J.E., Rusin, D., 2003. Efficient solution of rational conics. Math. Comput. 72 (243), 1417–1441 (electronic).

de Loera, J.A., Santos, F., 1996. An effective version of Pólya's theorem on positive definite forms. J. Pure Appl. Algebra 108 (3), 231–240.

Fukshansky, L., 2006. On effective Witt decomposition and Cartan-Dieudonné theorem, Canadian Journal of Mathematics (in press). Preprint http://www.math.tamu.edu/~lenny/witt.pdf.

Geddes, K.O., Czapor, S.R., Labahn, G., 1992. Algorithms for Computer Algebra. Kluwer Academic Publishers, Boston, MA.

Knebusch, M., 1976. On algebraic curves over real closed fields. II. Math. Z. 151 (2), 189–205.

Lam, T.Y., 2005. Introduction to quadratic forms over fields. In: Graduate Studies in Mathematics, vol. 67. American Mathematical Society, Providence, RI.

Powers, V., Wörmann, T., 1998. An algorithm for sums of squares of real polynomials. J. Pure Appl. Algebra 127 (1), 99–104.

Reznick, B., 2005. On the absence of uniform denominators in Hilbert's 17th problem. Proc. Amer. Math. Soc. 133 (10), 2829–2834 (electronic).

Scharlau, W., 1985. Quadratic and Hermitian forms. In: Grundlehren der Mathematischen Wissenschaften (Fundamental Principles of Mathematical Sciences), vol. 270. Springer-Verlag, Berlin.

Schweighofer, M., 2002. An algorithmic approach to Schmüdgen's Positivstellensatz. J. Pure Appl. Algebra 166 (3), 307–319.

Simon, D., 2005a. Quadratic equations in dimensions 4, 5 and more. Preprint http://www.math.unicaen.fr/~simon/maths/dim4.html.

Simon, D., 2005b. Solving quadratic equations using reduced unimodular quadratic forms. Math. Comput. 74 (251), 1531–1543 (electronic).

Szymiczek, K., 1997. Bilinear algebra. An introduction to the algebraic theory of quadratic forms. In: Algebra, Logic and Applications, vol. 7. Gordon and Breach Science Publishers, Amsterdam.

Witt, E., 1998. Collected Papers. Gesammelte Abhandlungen. Springer-Verlag, Berlin.

Wolfram Research, Inc., 1996, Mathematica, Version 3.01. Wolfram Research, Inc., Champaign, IL.